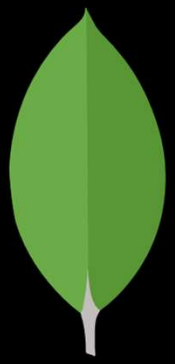

MongoDB

Introduction

Introduction to MongoDB

MongoDB is a popular NoSQL database known for its flexibility, scalability, and ease of use.

Its document-oriented structure makes it ideal for handling complex data, often found in modern applications.

MongoDB is known for its ability to store unstructured data and its support for dynamic schemas, making it particularly suitable for applications with rapidly changing data models.

Querying the Coffee Collection

1

Define the Collection
First, specify the coffee collection you want to query.

2

Construct the Query
Use MongoDB's query language to define criteria for filtering data.

3

Execute the Query
Send the query to MongoDB, which returns matching documents.

Let ‘s see

Performance Benefits

Indexing for Performance

1 Speed Up Queries

Indexes act like shortcuts, allowing MongoDB to quickly locate relevant documents.

2 Improve Scalability

Indexes help MongoDB handle large datasets more efficiently, efficiently, ensuring smooth performance.

3 Optimize Data Retrieval

Indexes enable MongoDB to retrieve specific data faster, reducing query execution time.

Filtering Data with Query Operators

1. Equality Operator (=)

   Find documents where a specific field matches a value.

2. Greater Than (>) and Less Than (<)

   Find documents where a field value falls within a specific range.

3. Regular Expressions

   Filter data based on pattern matching in strings, useful for searching coffee names or origins.

4. Logical Operators (AND, OR, NOT)

   Combine multiple conditions to refine search results, such as finding coffee beans both from Ethiopia and with a fruity flavor.

Aggregating Data with Pipelines

1

Stage 1: Match

Filter documents based on specified criteria, selecting only relevant coffees.

2

Stage 2: Group

Group documents based on a specific field, like coffee origin, to calculate statistics.

3

Stage 3: Sort

Order the grouped results based on a field, like the number of coffees per origin.

4

Stage 4: Limit

Restrict the final result set to a specific number of documents.

FIND QUERIES

- Count the no of customers in the doc.?
- db.customers.count()
  25

- Retrieve all the data where category is Latte?
- db.customers.find({category:"Latte"})

- Retrieve all the data where the order quantity is 3?
- db.customers.find({quantity:3})

- Retrieve all the data where the username contains the word “espresso”?
- db.customers.find({
- username:/espresso/})

- Retrieve all the data where the price range is more than 2 and the category is Mocha?
- db.customers.find({
- category:"Mocha",
- price:{"$gte":2}})

- Retrieve all the data where the category is Americano but not Latte?
- db.customers.find({
- category:"Americano",
- category:{"$ne":"Latte"}})

○ Find all the data where the id no is less than 20 and the product name starts from “M”? • db.customers.find({ • id:{"$lt":20}, • productname:/^M/})	• Retrieve all the data where the category is Latte and username should not contain “espresso”? • db.customers.find({ • category:"Latte", • username:{"$not":/espresso/}})	• Retrieve all the data where the username starts with “e” and ends with “t” • db.customers.find({ • username:{"$regex":"^e.*t$", • "$options":"i"}})
• Retrieve all the data where the username contains the word “espresso”? • db.customers.find({”$or": • [{username:/^espresso/}, {category:"Espresso"}]})	• Retrieve all the data where the quantity is between 1 and 3? • db.customers.find({ • quantity:{"$gte":1,"$lt":3}})	• Retrieve all the data where the category is Mocha quantity is 2 and price is 3 ? • db.customers.find({ • quantity:2, • category:" • Mocha",price:3})


AGGREGATION
QUERIES

- Group customers by category then find the average price of the product?

```
[{
$unwind:"$category"
},
{
$group: {
_id:"$category",
avgprice:{$avg:"$price"}
}
}
]
```

- Group customers by category then find the count of users?

```
[{
$unwind:"$category"
},
{
$group: {
_id:"$category",
countofuser:{$sum:1}
}
}
]
```

- Group customers by quantity and then find the maximum price of each quantity?

```
[
{
$group: {
_id:"$quantity",
maxprice:{$max:"$price}
}
}
]
```

- Group all user by price range and calculate the total no of user in each case ?

```
[
{
$bucket: {
groupBy: "$price",
boundaries: [1,2,3],
default: "other",
output: {count:{$sum:1},
username:{$push:
{
name:"$username"}}
}}}
]
```

- Group all user by price and collect all unique category for each price ?

```
[{$unwind:"$category"},
{ $group: {
_id:"$price",
skills:{$addToSet:"$category"},
users:{$push:"$$ROOT"}}
}
]
```

- Group all user by id and then find the count and average price of customers those have ordered Latte or Mocha ?

```
[{
$match: {

category:"Latte",category:"Mocha"
}
},
{
$group: {
_id:"$id",
countofuser:{$sum:1},
avgprice:{$avg:"$price"}
}}]
```

- Retrieve all the users that have same last name ?

```
[
{
$group: {
_id:"$username",
usercount:{$sum:1},
users:{$push:"$$ROOT"}}
}
]
```

- Group customers by quantity and then find the minimum price of each quantity?

```
[
{
$group: {
_id:"$quantity",
minprice:{$min:"$price"}
}
}
]
```

- Group all the users on the basis of category and create an array for each category?

```
[
{ $group: {
_id:"$category",
users:
{
$push:{
name:"$username"
}
}
}
}
]
```

- Group all user by price and collect all unique category for each price resetting the vale of each field?

```
[{$unwind:"$category"},

{ $group: {
 _id:"$price",
 category:
{$addToSet:"$category"},
 users:{$push:"$$ROOT"}}
},
{
$project: {
 _id:1,
category:"$category"
}
}]
```

- Group all user by id and then find count of user who have username espresso?

```
[
{
$match: {
username:/espresso/
}
},
{
$group: {
 _id:"$id",
countofuser:{$sum:1}
}
}
]
```

- Group all user by price and category then count the no of users where the quantity is less than 3 ?
- Group all user by id and then find count of user who have username espresso?

```
[
{
$match: {
username:/espresso/
}
},
{
$group: {
 _id:"$id",
countofuser:{$sum:1}
}}]
```

Conclusion :

MongoDB is a powerful and versatile database, offering a rich query language and features for managing data efficiently.

By understanding the concepts presented, you can confidently explore, manipulate, and analyze your coffee data in a seamless way.